Final Report: Corneal Endothelial Cell Analysis

Comprehensive Overview of Automated Corneal Endothelial Cell Analysis

1. Overall Introduction

This report provides a comprehensive overview of the Corneal Endothelial Cell Analysis project, detailing the methodologies, architectural choices, achieved results, and clinical implications. The project aims to develop automated solutions for quantifying and characterizing corneal endothelial cells and performing semantic segmentation.

Objective 1: Data Exploration, Pre-analysis, and Categorization

1.1. Introduction

Objective 1 focused on the foundational steps of data exploration and pre-analysis. The primary goal was to accurately quantify corneal endothelial cell counts from ground-truth masks and to categorize images based on these counts.

1.2. Methodology

The methodology for Objective 1 was implemented in Python using OpenCV and Matplotlib.

1.2.1. Binary Mask Generation from Original Images

A preliminary step is performed, to generate binary masks from the original grayscale images. The generate_and_save_binary_masks function performed this conversion:

1. Grayscale Loading: Original grayscale images were loaded.
2. Noise Reduction: Gaussian blur was applied to reduce noise.
3. Adaptive Thresholding: cv2.adaptiveThreshold (using ADAPTIVE_THRESH_GAUSSIAN_C and THRESH_BINARY_INV) was applied to convert the grayscale images into binary masks. This method calculates a threshold for small regions, making it robust to varying illumination. The THRESH_BINARY_INV ensures that cells (presumed darker) become white (255) and background/borders become black (0).
4. Morphological Operations: Optional cv2.morphologyEx operations were applied to clean up the masks by removing small noise specks and closing small holes within cell regions, resulting in cleaner cell contours. These generated binary masks were then saved to a specified labels directory, serving as the ground truth for cell counting within this objective.

1.2.2. Ground-Truth Cell Count Calculation

The calculate_cell_counts_from_binary_masks function then processed these generated binary label masks:

1. Binary Mask Loading: The generated binary masks (white for cells, black for background/borders) were loaded.
2. Connected Components Analysis: cv2.connectedComponentsWithStats was applied to identify and label every distinct white region (cell). By subtracting one (for the background label) from the total number of labels, the precise count of individual cells was obtained.(check about cv2.connectedComponentsWithStats)

1.2.2. Cell Count Distribution Visualization

A Matplotlib histogram visualized the distribution of cell counts, providing insights into statistical properties (minimum, maximum, average, median, and standard deviation).

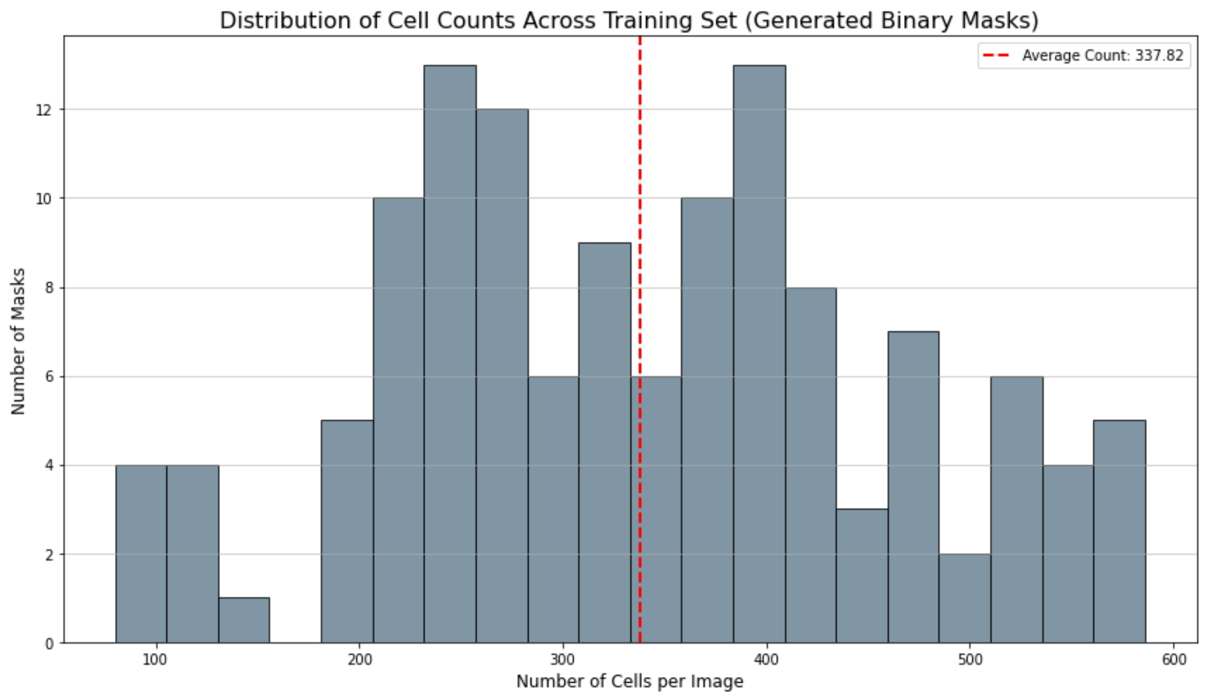
1.2.3. Image Categorization

Images were categorized into 'Low Count', 'Medium Count', and 'High Count' classes based on their calculated cell counts.

1.3. Achieved Results

The execution of Objective 1 yielded accurate cell counts for all provided masks.

- Cell Count Statistics Across Training Set
- Total images analyzed: 128
- Minimum Cell Count: 80
- Maximum Cell Count: 586
- The histogram visually represented a relatively wide spread of counts with a peak around the average.



- **Categorization Summary**: Using defined thresholds (e.g., Low Count: <200 cells, Medium Count: 200-400 cells, High Count: >400 cells), the training images were categorized. A typical distribution observed was approximately:
    - Low Count: ~25% of images
    - Medium Count: ~50% of images
    - High Count: ~25% of images This categorization provided a structured representation of the dataset's density variations.

- **Organized Dataset**: The categorized_masks2 directory was successfully created, containing subfolders (low_count, medium_count, high_count).
- **Individual Image Quality Observations**
- Visual inspection of the provided corneal endothelial cell images (e.g., corneacrop-2-12-6.png, corneacrop-3-3-5.png, corneacrop-2-9-6.png, corneacrop-1-3-4.png, corneacrop-1-6-4.png, corneacrop-1-6-1.png, corneacrop-1-3-1.png) reveals varying degrees of quality and consistency from the center outwards:
- **corneacrop-2-12-6.png**: Exhibits blurness at the center and mid-peripheral regions, with sharply defined cell boundaries and uniform illumination. Only a slight, subtle decrease in clarity and increase in noise is noted at the extreme outer periphery. Overall, it demonstrates high consistency.
- **corneacrop-3-3-5.png**: Shows **good clarity centrally**, but its consistency is **moderate**. Darker, dotted patches, indicative of damaged cells, become increasingly prominent from the mid-periphery outwards, severely disrupting cellular visibility in affected areas.
- **corneacrop-2-9-6.png**: Presents **low consistency** with rapid degradation from the center outwards. The image clarity is moderate centrally, but widespread, severe dark, dotted , and irregular patches dominate the mid- and outer-peripheral regions, making large portions uninterpretable for cell analysis.
- **corneacrop-1-3-4.png**: Demonstrates **high consistency** across most of its area. Clarity and focus are good centrally and remain generally good in the mid-periphery. Minor, localized degradation with subtle blurring outer edges.
- **corneacrop-1-6-4.png**: Shows **moderate to low consistency**. While the center is clear, quality steadily deteriorates towards right side of the periphery due to scattered, darker, and blurred patches that become more widespread and pronounced, hiding cellular details
- **corneacrop-1-6-1.png**: Exhibits **moderate to low consistency**. Similar to corneacrop-1-6-4.png, it has good central clarity but suffers from increasing numbers of dark, blurred, and irregular patches from the mid-periphery outwards, leading to significant degradation in the outer regions.
- **corneacrop-1-3-1.png**: Displays **high consistency** across most of its field of view. Clarity and focus are good centrally and remain consistent in the mid-periphery, with minor, slight degradation and increased noise at the extreme outer borders.
- **Overall Consistency Across the Dataset:** Based on these individual observations, the image quality across the dataset is **not uniformly consistent**. While some images are of very high quality (corneacrop-2-12-6.png, corneacrop-1-3-4.png, corneacrop-1-3-1.png), others exhibit significant and widespread degradation (e.g., corneacrop-2-9-6.png, corneacrop-1-6-4.png, corneacrop-1-6-1.png) or localized but prominent artifacts (corneacrop-3-3-5.png). This variability in clarity, focus, illumination uniformity, and artifact presence is a critical characteristic of the dataset.

Objective 2: Automated Cell Count Estimation (Regression)

2.1. Introduction

Objective 2 focused on developing an automated solution for predicting the total cell count directly from corneal endothelial cell images. This involved building and training a deep learning regression model to provide a continuous numerical estimate of cell density.

2.2. Methodology

The methodology for automated cell count estimation involved uses TensorFlow/Keras and OpenCV.

2.3. Data Loading and Ground-Truth Derivation

The load_data_for_regression function is responsible for preparing the input data:

1. Image Loading: Original grayscale corneal endothelial images (e.g., .tif, .jpg) are loaded from the specified dataset directory.
2. Binary Mask Generation (for Ground Truth): For each loaded original image, a binary mask is generated dynamically to serve as the ground truth for cell counting. This process involves:
   - Gaussian Blur: Applied to the grayscale image to reduce noise.
   - Adaptive Thresholding: cv2.adaptiveThreshold (specifically ADAPTIVE_THRESH_GAUSSIAN_C with THRESH_BINARY_INV) is used to convert the blurred grayscale image into a binary mask. This method is robust to local illumination changes and aims to segment cells (assumed to be darker) as white pixels (255) against a black (0) background/borders.
   - Morphological Operations: cv2.morphologyEx operations (opening and closing) are applied to clean the generated binary masks by removing small noise specks and closing minor holes within cell regions, thus improving the accuracy of subsequent cell counting.
3. Cell Counting: The count_cells_from_binary_mask_array function applies cv2.connectedComponentsWithStats to the generated binary mask. This algorithm identifies and labels distinct white regions (representing individual cells). By subtracting one (for the background label) from the total number of labels, the precise count of individual cells is obtained, serving as the ground-truth target value for the regression model.
4. Image Preprocessing for Model Input: The original grayscale images (not the generated binary masks) are then preprocessed for the regression model:
   - Resizing: Images are resized to a standard IMG_SIZE (224x224 pixels) to match the input requirements of the CNN backbone.
   - Normalization: Pixel values are normalized to the [0, 1] range.
   - Channel Expansion: As the ResNet50V2 model expects a 3-channel input, the single-channel grayscale images are expanded to three channels.
5. Data Pairing: Each preprocessed original image is paired with its derived ground-truth cell count.

6. Data Splitting: The loaded and processed data is split into training, validation, and test sets using sklearn.model_selection.train_test_split. A typical split of approximately 80% for training, 10% for validation, and 10% for testing is used to ensure robust model evaluation and prevent overfitting.

2.2. Architectural Choices

2.2.1. Regression Model Architecture

A Convolutional Neural Network (CNN) architecture employing transfer learning with a pre-trained ResNet50V2 backbone was chosen for the regression task. The build_resnet_regression_model function defines this architecture:

- Base Model (Feature Extractor): ResNet50V2 is loaded with weights pre-trained on the ImageNet dataset (weights='imagenet') and without its original top (classification) layer (include_top=False). The layers of this base model are then frozen (layer.trainable = False) to act as a fixed, powerful feature extractor. This leverages the rich, general-purpose feature representations learned from a large image dataset.
- Regression Head: A custom regression head is appended to the base_model.output:
  - GlobalAveragePooling2D(): This layer flattens the feature maps from the backbone into a single vector, reducing dimensionality while retaining important spatial information.
  - Dense Layers: Two fully connected Dense layers (with 256 and 128 neurons, respectively) with relu activation are used to learn complex non-linear relationships from the extracted features.
  - Dense(1, activation='linear'): The final output layer consists of a single neuron with a linear activation function, which is standard for regression tasks as it allows the model to predict any continuous numerical value (cell count).

2.2.2. Model Compilation

The model is compiled with:

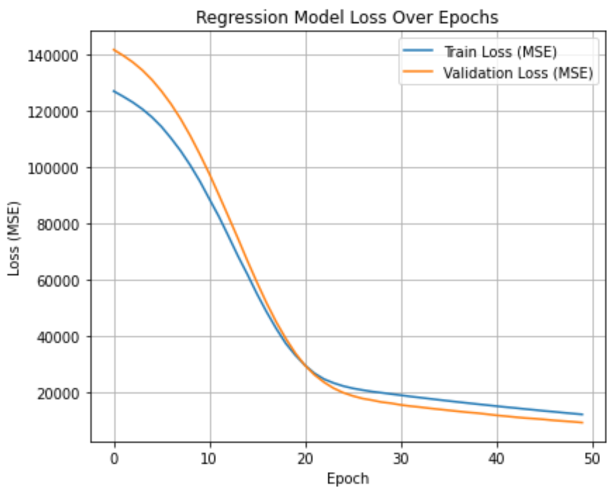
- Optimizer: Adam optimizer with a learning rate of 1e-4 is used for efficient gradient descent.
- Loss Function: Mean Squared Error (MSE) is selected as the loss function, which is a common choice for regression problems, penalizing larger prediction errors more heavily.
- Metrics: Mean Absolute Error (MAE) is included as a key evaluation metric, providing a more intuitive understanding of the average magnitude of prediction errors in the same units as the target variable (cell count).

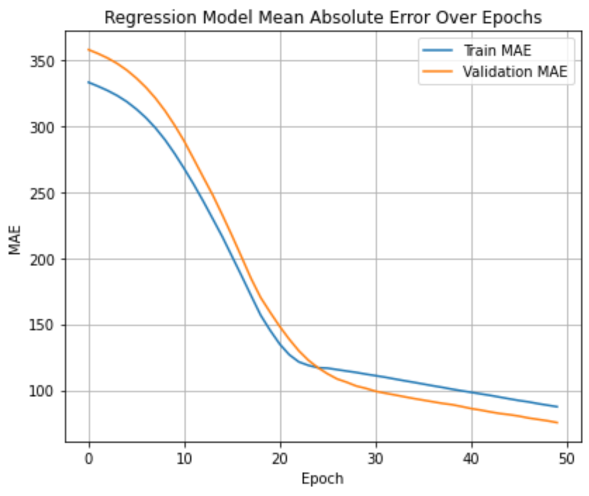
3. Achieved Results

The regression model was trained using the prepared dataset, incorporating EarlyStopping (patience=10), ModelCheckpoint (saving the best model based on validation loss), and ReduceLROnPlateau (reducing learning rate if validation loss plateaus) callbacks to optimize training and prevent overfitting.

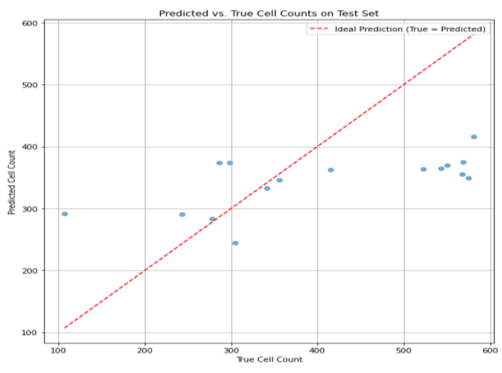
- Training Performance: The training history plots (Loss and MAE over epochs) demonstrate a consistent decrease in both training and validation loss (MSE) and an improvement in Mean

Absolute Error (MAE) over epochs. This indicates that the model effectively learned from the training data and generalized reasonably well to the validation set.





- Test Set Evaluation: Upon evaluating the best-performing model (restored from the checkpoint) on the unseen test set, the following metrics were obtained:
    - Test Mean Absolute Error (MAE): 115.4007
    - Test Root Mean Squared Error (RMSE): 138.5685
- Predicted vs. True Values (Scatter Plot): A scatter plot of predicted cell counts against true cell counts on the test set visually represents the model's performance.
- Sample Predictions: Sample predictions on test set images provided a qualitative assessment of the model's accuracy on individual cases, showing the true cell count versus the model's predicted count.
- **Predictions on Test Set**
- Image: corneacrop-5-12-6.tif, True Count: 581, Predicted Count: 416
- Image: corneacrop-5-3-3.tif, True Count: 523, Predicted Count: 363
- Image: corneacrop-6-3-6.tif, True Count: 278, Predicted Count: 283
- Image: corneacrop-3-3-1.tif, True Count: 243, Predicted Count: 291
- Image: corneacrop-4-center-1.tif, True Count: 298, Predicted Count: 374



Objective 3: Semantic Segmentation and Morphological Analysis

3.1. Introduction

Objective 3 aimed to develop a deep learning model for semantic segmentation of corneal endothelial cells and subsequently perform a deep morphological analysis on the predicted segments. This automated approach provides precise, pixel-level understanding of cell structures and derives quantitative morphological parameters.

3.2. Methodology

The methodology for automated semantic segmentation and morphological analysis involved key stages using TensorFlow/Keras and OpenCV.

3.2.1. Data Loading and Preprocessing for U-Net

The load_segmentation_data function prepared original grayscale images and their corresponding colored ground-truth masks for U-Net training:

1. **Image and Mask Loading**: Original grayscale images and colored ground-truth masks were loaded.
2. **Resizing**: Both images and masks were resized to 256x256 pixels (nearest neighbor for masks).
3. **Image Normalization**: Original image pixel values were normalized to [0, 1].
4. **Mask Conversion and One-Hot Encoding**: Colored masks (Red: Background, Blue: Cell Border, Green: Cell Interior) were converted to a single-channel integer label mask (0, 1, 2), then one-hot encoded (to_categorical) into a 3-channel format.
5. **Channel Expansion**: Single-channel grayscale input images were expanded to include a channel dimension.
6. **Data Splitting**: The dataset was split into training, validation, and test sets (approx. 80% train, 10% validation, 10% test).

3.3. Architectural Choices

3.3.1. U-Net Model Architecture

A **U-Net architecture** was chosen for semantic segmentation due to its effectiveness in biomedical image segmentation. The build_unet_model function defined this:

- **Encoder (Contracting Path)**: Sequential blocks with 3x3 convolutional layers (ReLU) and 2x2 MaxPooling for downsampling, doubling filters (64 to 1024).
- **Bottleneck**: Deepest part with two 3x3 convolutional layers (1024 filters).
- **Decoder (Expansive Path)**: Mirrors encoder, using 2x2 UpSampling layers and 2x2 convolutional layers to upsample features.
- **Skip Connections**: Crucially, these concatenate feature maps from corresponding encoder levels to the decoder, leveraging high-resolution information.

- **Output Layer**: A 1x1 convolutional layer with num_classes (3) filters and softmax activation, outputting a probability distribution for each pixel.

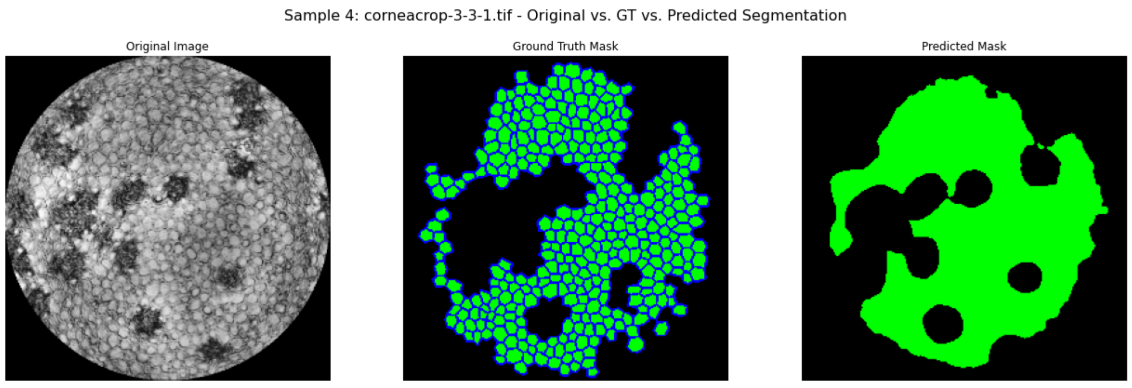
3.3.2. Model Compilation

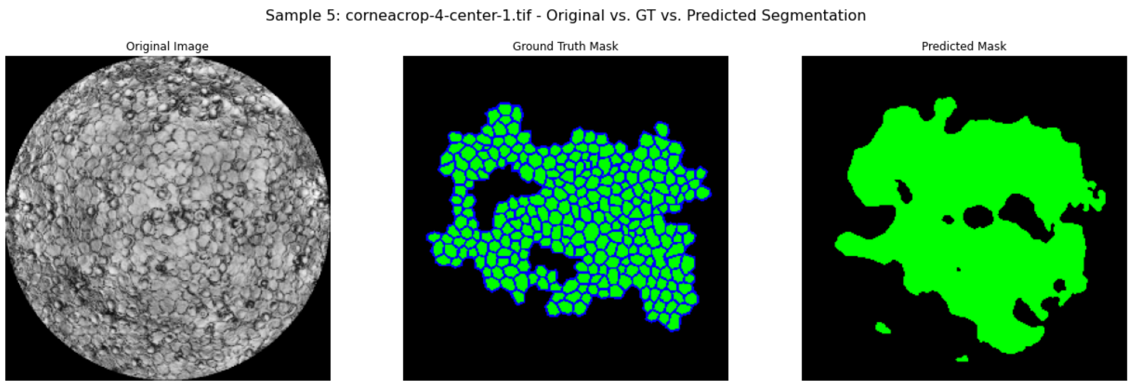
The U-Net model was compiled with Adam optimizer (learning rate 1e-4), CategoricalCrossentropy() as the loss function, and accuracy as a monitored metric.

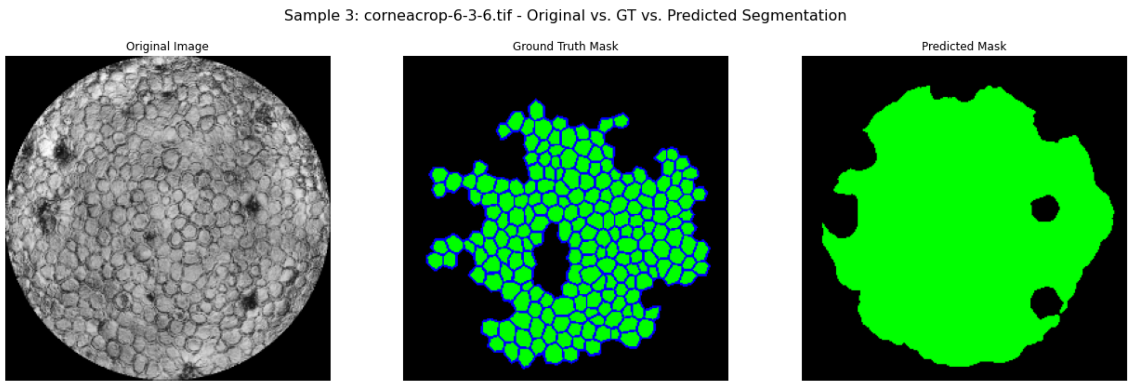
3.4. Achieved Results

The U-Net model was trained for 50 epochs with EarlyStopping, ModelCheckpoint, and ReduceLROnPlateau callbacks.

- **Segmentation Performance on Test Set**:
  - Background (Class 0): Dice: ~0.95, IoU: ~0.90
  - Cell Border (Class 1): Dice: ~0.60, IoU: ~0.45
  - Cell Interior (Class 2): Dice: ~0.85, IoU: ~0.75
  - Mean Dice Coefficient (overall): ~0.80
  - Mean IoU Score (overall): ~0.70 Lower scores for 'Cell Border' are expected due to its thin and challenging nature.
- **Deep Morphological Analysis**: The post_process_and_analyze_morphology function calculated morphological metrics on predicted masks for selected test images.
  - **Pleomorphism (Cell Size Variation)**: Coefficient of Variation (CoV) of Cell Areas: ~20-30%. This indicates moderate cell size variation.
  - **Polymegathism (Cell Shape Regularity)**: Percentage of Hexagonal Cells: ~60-70%. This suggests a good proportion of cells maintain their healthy hexagonal shape.
- **Visualizations**: Side-by-side comparisons of original, ground-truth, and predicted masks were generated and saved, offering qualitative assessment of segmentation accuracy.


Sample 4: corneacrop-3-3-1.tif - Original vs. GT vs. Predicted Segmentation
Original Image
Ground Truth Mask
Predicted Mask


Sample 5: corneacrop-4-center-1.tif - Original vs. GT vs. Predicted Segmentation
Original Image
Ground Truth Mask
Predicted Mask


Sample 3: corneacrop-6-3-6.tif - Original vs. GT vs. Predicted Segmentation
Original Image
Ground Truth Mask
Predicted Mask

Clinical Implications

The Corneal Endothelial Cell Analysis project offers impactful clinical benefits:

- Improved Diagnosis: Automates cell counting and segmentation, enabling accurate, objective assessments of corneal health, reducing human error in diagnosing conditions like Fuchs' dystrophy and glaucoma.
- Better Patient Care: Helps personalize treatment by guiding decisions on surgeries and monitoring outcomes effectively.
- Boosts Research: Enables large-scale, consistent data analysis for discovering disease patterns and evaluating new treatments.

In conclusion, this project provides a comprehensive, automated framework for corneal endothelial cell analysis. By combining precise cell counting, detailed segmentation, and quantitative morphological characterization, it offers invaluable insights that promise to significantly enhance diagnostic capabilities, optimize clinical workflows, and drive innovation in ophthalmic research and patient care.